2020 Fall - ECE 2220 Laboratory 4 : A 4-1 Multiplexer

Instructions

1) Draw the internal logic circuit diagram of a 4-1 multiplexer using AND, OR and NOT gates and include it in your report.

2) Implement the this circuit using AND, OR and NOT gates in Verilog.

a) Write your Verilog code and upload the file.

b) Run a simulation of you Verilog code of AND_OR 4-1 mux and add it to your report.

c) Use switches SW[0], SW[1], SW[2] and SW[3] as the mux inputs. Use SW[8] and SW[9] as your selects. For the select values switches use switched SW[8]-SW[9]. Use the corresponding LEDs for the inputs and selects (i.e SW[0]-SW[3] go to - LEDR[0]- LEDR[3]) and SW[8]-SW[9] go to LED[8] -LED[9]. Put the output MUX value (f) on LED[6].

3) Implement the same circuit this time using a "case" statement to model the MUX.

a) Write your Verilog code, upload the file.

b) Run a simulation of you Verilog code of AND_OR 4-1 mux and add it to your report.

c) Use switches SW[0], SW[1], SW[2] and SW[3] as the mux inputs. Use SW[8] and SW[9] as your selects. For the select values switches use switched SW[8]-SW[9]. Use the corresponding LEDs for the inputs and selects (i.e SW[0]-SW[3] go to - LEDR[0]- LEDR[3]) and SW[8]-SW[9] go to LED[8] -LED[9]. Put the output MUX value (f) on LED[6].

4) Implement the same circuit this time using the selects as an index to model the MUX.

a) Write your Verilog code, upload the file.

b) Run a simulation of you Verilog code of AND_OR 4-1 mux and add it to your report.

c) Use switches SW[0], SW[1], SW[2] and SW[3] as the mux inputs. Use SW[8] and SW[9] as your selects. For the select values switches use switched SW[8]-SW[9]. Use the corresponding LEDs

5) Comment on the complexity and the expandability of the two Verilog codes (primitives or case statement) to implement the same MUX. Is there an even simpler way to implement this MUX?